Introduction and Background for Seeing with Sound

Introduction

Seeing with sound is our attempt to meaningfully transform an image to sound. The motivation behind it is simple, to convey visual information to blind people using their sense of hearing. We believe in time, the human brain can adapt to the sounds, making it a useful and worthwhile system.

Background and Problems

In researching for this project, we found one marketed product online, the, vOICe, that did just what we set out to do. However, we believe that the vOICe is not optimum, and we have a few improvements in mind. One idea is to make the center of the image the focus of the final sound. We feel like the center of an image contains the most important information, and it gets lost in the left to right sweeping of vOICe. Also, some of the images are far too "busy" to use their technique. We the images need to be simplified so that only the most important information is conveyed in the sounds.

Seeing using Sound - Design Overview

Input Filtering

The first step in our process is to filter the input image. This process helps solve the "busy" sound problem from the vOICe. We decided to first smooth the image with a low pass filter, leaving only the most prominent features of the image behind. We then wanted to filter the result with an edge detector, essentially a high pass filter of some sort. We chose to use a Canny filter for the edge detection. The advantage of using an edge detector lies in simplifying the image while at the same time highlighting the most structurally significant components of an image. This is especially applicable to using the system for the blind, as the structural features of the image are the most important to find your way around a room.

The Mapping Process

Simply put, the mapping process is the actual transformation between visual information and sound. This block takes the data from the filtered input, and produces a sequence of notes representing the image. The process of mapping images to sound is a matter of interpretation, there is no known "optimal" solution to the mapping for the human brain. Thus, we simply chose an interpretation that made sense to us.

First of all, it seemed clear to us that the most intuitive use of frequency would be to correlate it to the relative vertical position of an edge in the picture. That is, higher frequencies should correspond to edges that are higher in the image than lower frequencies. The only other idea that we wanted to stick to was making the center the focus of the attention. For a complete description of this component, see the mapping process.

Canny Edge Detection

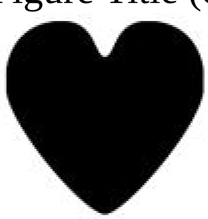
Introduction to Edge Detection

Edge detection is the process of finding sharp contrasts in intensities in an image. This process significantly reduces the amount of data in the image, while preserving the most important structural features of that image. Canny Edge Detection is considered to be the ideal edge detection algorithm for images that are corrupted with white noise. For a more in depth introduction, see the [Canny Edge Detection Tutorial].

Canny Edge Detection and Seeing Using Sound

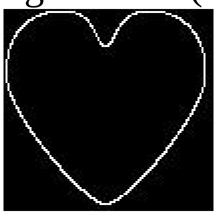
The Canny Edge Detector worked like a charm for Seeing Using Sound. We used a Matlab implementation of the Canny Edge Detector, which can be found at http://ai.stanford.edu/~mitul/cs223b/canny.m. Here is an example of the results of filtering an image with a Canny Edge Detector:

Figure Title (optional)



Before Edge Detection

Figure Title (optional)



After Edge Detection

Seeing using Sound's Mapping Algorithm

The mapping algorithm is the piece of the system that takes in an edge-detected image, and produces a sound clip representing the image. The mapping as we implemented it takes three steps:

- Vertical Mapping
- Horizontal Mapping
- Color Mapping

Mapping Diagram

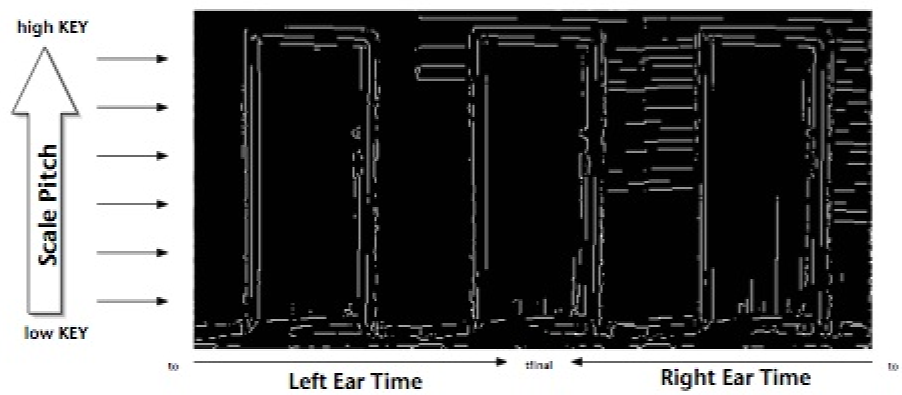


Illustration of our mapping algorithm

Vertical Mapping

The first step of the algorithm is to map the vertical axis of the image to the frequency content of the output sound at a given time. We implemented this by having the relative pitch of the output at that time correspond to rows in each column that have an edge. Basically, the higher the note you hear, the higher it is in your field of vision, and the lower the note, the lower in your field of vision.

Horizontal Mapping

Next, we need some way of mapping the horizontal axis to the output sound. We chose to implement this by having our system "sweep" the image from the outside-in in time (see figure 1). The reasoning behind this is that the focus of the final sound should be the center of the field of vision, so we have everything meeting in the middle. This means that each image will have some period that it will take to be "displayed" as sound. The period begins at some time t0, and, with stereo sound, the left and right channels start sounding notes corresponding to edges on each side of the image, finally meeting in the middle at some time tf.

Color Mapping

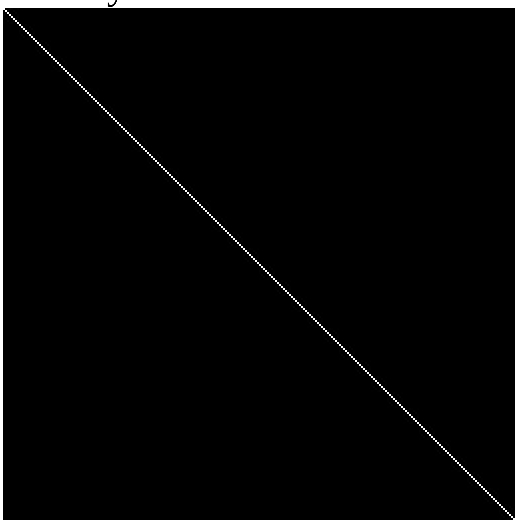
Using scales instead of continuous frequencies for the notes gives us some extra information to work with. We decided to also try to incorporate the color from the original image of the point at an edge. We were able to do this by letting the brightness of the scale that we use. For example, major scales sound much brighter than minor scales, so bright colors correspond to major scales, and darker ones correspond to minor. This effect is difficult to perceive for those that aren't trained, but we believe that the brain can adapt to this pattern regardless of whether or not the user truly understands the mapping.

Demonstrations of Seeing using Sound

For each example, right click on the link to the corresponding sound and go to "Save Link Target As..." to download and play it.

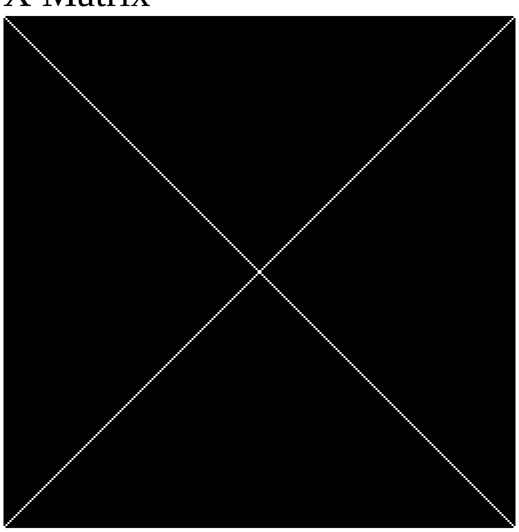
Examples

Identity Matrix



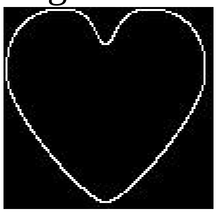
Our Simplest Example - Listen

X Matrix



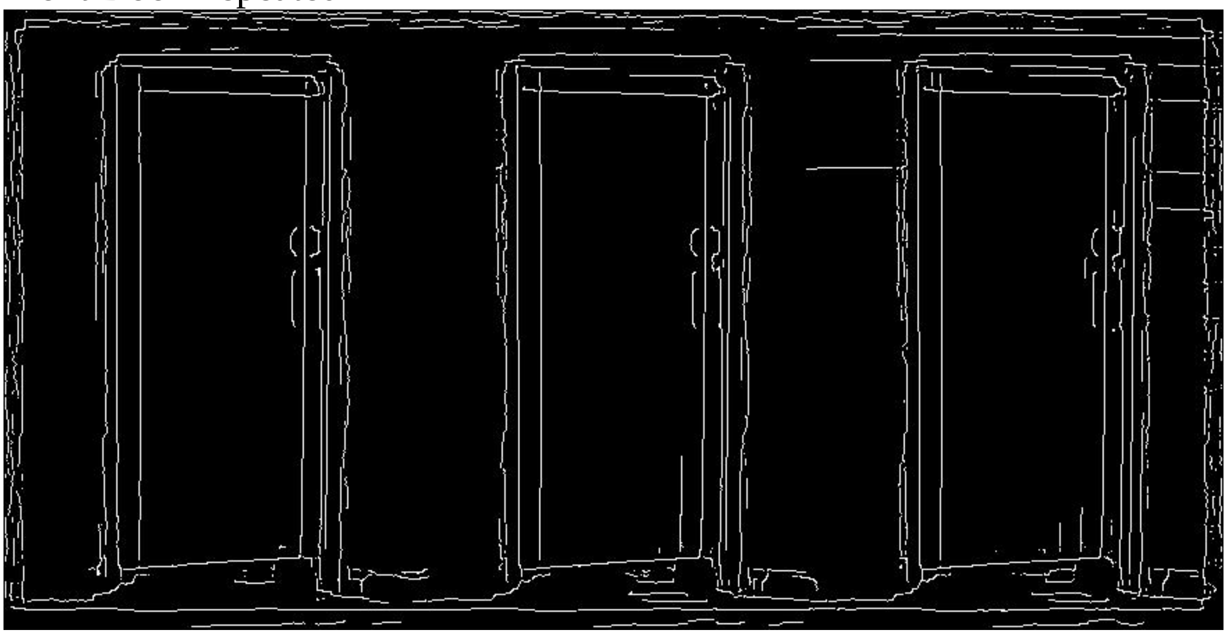
Listen

Edge Detected Heart



Listen

Front Door Repeated



Our Hardest Example - Not for beginners! - Listen

Final Remarks on Seeing using Sound

Future Considerations and Conclusions

There are many ways to improve upon our approach. One way to significantly improve left/right positioning is to have the left and right scales play different instruments. Another way to improve resolution would be to have different neighboring blocks compare data so that when an edge spans many different blocks it does not sound like a cacophony. Other filters could be applied, besides edge detectors, to determine other features of the image, such as color gradients or the elements in the foreground. This information could be encoded into different elements of the basis scale, or even change the scale to a different, perhaps acyclic, pattern. One way to go about this might be to look at existing photo processing filters (e.g. in Photoshop) and use those for inspiration.

Contact Information of Group Members

- **Flatow, Jared**: jmflizz @ rice dot edu
- **Hall, Richard**: rlhall @ rice dot edu
- **Shepard, Clay**: cwshep @ rice dot edu